



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/670,826

09/25/2003

Barry L. Minor

AUS920030790US1

7316

40412 7590 01/15/2008
IBM CORPORATION- AUSTIN (JVL)
C/O VAN LEEUWEN & VAN LEEUWEN
PO BOX 90609
AUSTIN, TX 78709-0609

EXAMINER

TRUONG, CAMQUY

ART UNIT

PAPER NUMBER

2195

MAIL DATE

DELIVERY MODE

01/15/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/670,826	Applicant(s) MINOR ET AL.	
	Examiner Camquy Truong	Art Unit 2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE ____ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 02 November 2007.
- 2a) ☒ This action is FINAL. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,3-8,10-14 and 16-20 is/are pending in the application.
- 4a) Of the above claim(s) ____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) ____ is/are allowed.
- 6) ☒ Claim(s) 1,3-8,10-14, 16-20 is/are rejected.
- 7) ☐ Claim(s) ____ is/are objected to.
- 8) ☐ Claim(s) ____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on ____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. ____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|--|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date: ____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date <u>10/2/07, 12/27/07</u> . | 6) <input type="checkbox"/> Other: ____ |

DETAILED ACTION

1. Claims 1, 3-8, 10-14, and 16-20 are presented for examination. Claims 2,9,15 have been cancelled.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

3. Claims 1, 3, 5-7, 14-16, 19-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ansari et al. (U.S. Patent 6,473,897) in view of Civlin (U.S. Patent Publication 20050028148).

4. As to claims 1 and 14, Ansari teaches a method for load balancing code execution, said method comprising:

compiling source code, the compiling resulting in byte code (assembly code generated by a compiler from the program code, col. 9, lines 12-14);

in response to retrieving the byte code, using the runtime loader to identify a processor type from a plurality of heterogeneous processor types to execute the byte code (the assembly code provides a series of tests for processor types. The test are performed during execution time, col. 9, lines 14-22; col. 8, lines 45-51; col. 13, lines 13-16, lines 30-33);

in response to identifying the processor type, using the runtime loader to translate the byte code to object code (the assembly code is subsequently converted to object code which is executed by processor, col. 10, lines 10-14); and loading the object code into a processor that corresponds to the identified processor type (the code checks with test whether the processor type is a Petium III processor. If it is the transferring program execution the the Pentium III processor, col. 9, lines 23-34; col. 13, lines 15-20, and lines 33-38).

5. Ansari does not explicitly teach retrieving the byte code at runtime using a runtime loader. However, Civlin teaches retrieving the byte code at runtime using a runtime loader (obtain byte code for executed a new hardware architecture, paragraph 25).

6. It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching retrieving the byte code at runtime using a runtime loader as taught by Civlin to the invention of Ansari because this allows the new hardware architecture executes the byte code without the portions of the code inevitably are lost.

7. As to claims 3 and 16, Ansari teaches the byte code includes a byte code type, the byte code type selected from the group consisting of Java, XML, HTML, Shader, and Script (col. 15, lines 1-5).

8. As to claim 5, Ansari teaches the identifying includes using the runtime loader to analyze analyzing the availability of each of the plurality of processor types, and wherein the analyzing includes retrieving a loading factor for each of the plurality of processor types which corresponds to the availability of each of the plurality of heterogeneous processor type (col. 8, lines 41-51).

9. As to claims 6-7 and 19-20, Ansari teaches:
detecting, using the runtime loader at runtime, one or more operations included in the byte code (col. 5, lines 15-22; col. 6, lines 10-19); and

Art Unit: 2195

matching, using the runtime loader at runtime, one or more of the operations with one of the processor types from the plurality of processor types (col. 5, lines 23-32; col. 6, lines 20-31).

10. Claims 4 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ansari et al. (U.S. Patent 6,473,897) in view of Civlin (U.S. Patent Publication 20050028148), and further in view of Kempf et al. (U.S. Patent 5359721).

11. As to claims 4, and 17, Ansari and Civlin do not explicitly teach determining whether to store a pointer in a byte code file, the pointer including a stored location that corresponds to the byte code;

storing the pointer in the byte code file in response to the determination; storing the byte code at the stored location in response to the determination; and performing the retrieving using the pointer, wherein the retrieving includes analyzing the stored location and retrieving the byte code in response to the analyzing.

However, Kempf does not determining whether to store a pointer in a byte code file, the pointer including a stored location that corresponds to the byte code (col. 7, lines 26-28);

storing the pointer in the byte code file in response to the determination; storing the byte code at the stored location in response to the determination (col. 8, lines 31-56); and

performing the retrieving using the pointer, wherein the retrieving includes analyzing the stored location and retrieving the byte code in response to the analyzing (col. 3, lines 14-22).

12. It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of determining whether to store a pointer in a byte code file, the pointer including a stored location that corresponds to the byte code; storing the pointer in the byte code file in response to the determination; storing the byte code at the stored location in response to the determination; and performing the retrieving using the pointer, wherein the retrieving includes analyzing the stored location and retrieving the byte code in response to the analyzing as taught by Kempf to the invention of Ansari and Civlin because this allows a process executing in non-supervisor mode to perform dynamic linking across address spaces with the program code without compromising system security.

13. **Claims 8, 10, 12-13, and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ansari et al. (U.S. Patent 6,473,897) in view of Civlin (U.S. Patent Publication 20050028148), and further in view of Jacobson (U.S. Patent Application Publication 2003/0188045).**

14. As to claim 8, Ansari teaches a method for load balancing code execution, said method comprising:

a plurality of processors (col. 5, lines 15-32);

a memory accessible by the processors (col. 9, lines 12-33; col. 10, lines 42-50);

compiling source code, the compiling resulting in byte code (assembly code generated by a compiler from the program code, col. 9, lines 12-14);

in response to retrieving the byte code, using the runtime loader to identify a processor type from a plurality of heterogeneous processor types to execute the byte code (the assembly code provides a series of tests for processor types. The test are performed during execution time, col. 9, lines 14-22; col. 8, lines 45-51; col. 13, lines 13-16, lines 30-33);

in response to identifying the processor type, using the runtime loader to translate the byte code to object code (the assembly code is subsequently converted to object code which is executed by processor, col. 10. lines 10-14); and
loading the object code into a processor that corresponds to the identified processor type (the code checks with test whether the processor type is a Petium III processor. If it is the transferring program execution the the Pentium III processor, col. 9, lines 23-34; col. 13, lines 15-20, and lines 33-38).

15. Ansari does not explicitly teach retrieving the byte code at runtime using a runtime loader. However, Civlin teaches retrieving the byte code at runtime using a runtime loader (obtain byte code for executed a new hardware architecture, paragraph 25).

Art Unit: 2195

16. It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching retrieving the byte code at runtime using a runtime loader as taught by Civlin to the invention of Ansari because this allows the new hardware architecture execute byte code without portions of the code inevitably are lost.

17. Ansari and Civlin do not explicitly teach a code execution load-balancing tool for load balancing code execution, the code execution load-balancing tool. However, Jacobson teaches a code execution load-balancing tool for load balancing code execution, the code execution load-balancing tool (paragraph 6, lines 13-15; paragraph 7, lines 1-4; paragraph 29, lines 3-7).

18. It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of code execution load-balancing tool for load balancing code execution, the code execution load-balancing tool as taught by Jacobson to the invention of Ansari and Civlin because this allows to balance workload tasks among multiple processors to fully utilize the resources associated with each processor.

19. As to claim 10, Ansari teaches the byte code includes a byte code type, the byte code type selected from the group consisting of Java, XML, HTML, Shader, and Script (col. 15, lines 1-5).

20. As to claims 12-13, Ansari teaches:

Detecting, using the runtime loader at runtime, one or more operations included in the code segment (col. 5, lines 15-22; col. 6, lines 10-19); and

Matching, using the runtime loader at runtime, one or more of the operations with one of the processor types from the plurality of processor types (col. 5, lines 23-32; col. 6, lines 20-31).

21. As to claim 18, Ansari teaches the identifying includes using the runtime loader to analyze analyzing the availability of each of the plurality of processor types, and wherein the analyzing includes retrieving a loading factor for each of the plurality of processor types which corresponds to the availability of each of the plurality of heterogeneous processor type (col. 8, lines 41-51).

22. Claim 11 is rejected under 35 U.S.C. 103(a) as as being unpatentable over Ansari et al. (U.S. Patent 6,473,897) in view.of Civlin (U.S. Patent Publication 20050028148), further in view of Jacobson (U.S. Patent Application Publication 2003/0188045), and further in view of Kempf et al. (U.S. Patent 5359721).

23. As to claim 11, Ansari, Civlin, and Jacobson do not explicitly teach determining whether to store a pointer in a byte code file, the pointer including a stored location that

Art Unit: 2195

corresponds to the byte code; storing the pointer in the byte code file in response to the determination; storing the byte code at the stored location in response to the determination; and performing the retrieving using the pointer, wherein the retrieving includes analyzing the stored location and retrieving the byte code in response to the analyzing.

24. However, Kempf does not determining whether to store a pointer in a byte code file, the pointer including a stored location that corresponds to the byte code (col. 7, lines 26-28);

storing the pointer in the byte code file in response to the determination; storing the byte code at the stored location in response to the determination (col. 8, lines 31- 56); and performing the retrieving using the pointer, wherein the retrieving includes analyzing the stored location and retrieving the byte code in response to the analyzing (col. 3, lines 14-22).

25. It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of determining whether to store a pointer in a byte code file, the pointer including a stored location that corresponds to the byte code; storing the pointer in the byte code file in response to the determination; storing the byte code at the stored location in response to the determination; and performing the retrieving using the pointer, wherein the retrieving includes analyzing the stored location and retrieving the byte code in response to the analyzing as taught by

Art Unit: 2195

Kempf to the invention of Ansari, Civlin, and Jacobson because this allows a process executing in non-supervisor mode to perform dynamic linking across address spaces with the program code without compromising system security.

Response to the argument

26. Applicant's arguments filed 11/2/07 for claims 1, 3-8, 10-14, and 16-20 have been considered but are moot in view of the new ground(s) rejection.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

Art Unit: 2195

Conclusion

27. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Camquy Truong whose telephone number is (571) 272-3773. The examiner can normally be reached on 8AM – 5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-3756.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIP. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIP system, contact the Electronic Business Center (EBC) at 866-217-9197(toll-free).

Camquy Truong

January 12, 2008


MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100